

PROTÓTIPO DE APLICAÇÃO DE PROGRAMAÇÃO VISUAL PARA CONTROLE DE MECANISMOS E SENSORES COM RASPBERRY-PI E NODE.JS

Eduarth Heinen¹, André George Lauer², Pedro Lealdino Filho³

¹ Aluno do curso de TSI da UTFPR-Guarapuava

² Prof. (titulação) do curso de TSI da UTFPR-Guarapuava
eduartheinen@gmail.com, andrelauer@utfpr.edu.br

Resumo

Considerando os resultados positivos obtidos por experimentos utilizando conceitos de robótica em lições de programação, esse projeto propõe o desenvolvimento de uma ferramenta que estimule a exploração da programação e da robótica, oferecendo uma interface de programação visual acessível capaz de controlar componentes – como leds, botões, sensores e atuadores – conectados ao minicomputador Raspberry Pi.

1. Introdução

A popularização de ferramentas como Raspberry Pi[1] e Arduino¹ tem estimulado a exploração de aplicações da robótica na educação, como demonstram os experimentos realizados em Faro[2] e em escolas na Inglaterra e País de Gales[3]. Nesses experimentos, resultados positivos foram obtidos utilizando robôs para estimular o raciocínio matemático e a dedução lógica.

Da revisão da literatura percebe-se também o interesse dos pesquisadores em estimular o surgimento de novos talentos da programação e engenharia. Interesse compartilhado por organizações como Code.org[4] e CAS - Computing at School[5], que trabalham pela inserção da ciência da computação no currículo escolar e pela popularização da programação.

Neste sentido, este projeto propõe o desenvolvimento de uma aplicação que permita conectar um conjunto de componentes – leds, botões, atuadores e sensores – ao Raspberry Pi e programar seu comportamento por meio de uma interface gráfica, arrastando e organizando elementos visuais que representem cada componente na interface. Dessa forma pretende-se construir uma ferramenta aberta e acessível para principiantes voltada para a exploração da programação e da robótica.

2. Materiais e Métodos

O projeto consiste em um minicomputador Raspberry Pi como *host* de uma aplicação Node.js e *controlador* de um conjunto de componentes – como leds, botões, sensores, motores – representados na interface da aplicação como elementos visuais da biblioteca Blockly[7].

Por ser baseada em Raspberry Pi a aplicação deverá ser capaz de controlar as portas físicas (GPIO) e responder a eventos de sensores e botões através da

¹ Arduino homepage: <http://arduino.cc/>

interface entre Node.js e hardware, criada pela biblioteca Onoff[8] e o sistema operacional Raspbian[9].

Utilizando o editor Blockly o usuário pode criar rotinas lógicas arrastando blocos coloridos e definindo valores, como mostra a figura 1. É possível também definir novos blocos, representando conceitos específicos da aplicação, como por exemplo o movimento de um personagem em um jogo.

A princípio a interface deve oferecer blocos que representem estruturas básicas de controle como *if-elses*, *whiles*, *fors*, blocos específicos para representar as portas físicas do Raspberry Pi e seu estado, e blocos que representem a instrução *wait*. O usuário deve então ser capaz de programar uma sequência de ações executadas pelos componentes conectados ao controlador. A interface deve oferecer um botão executar, que interprete o código javascript gerado pela biblioteca Blockly e execute as instruções do lado do controlador.

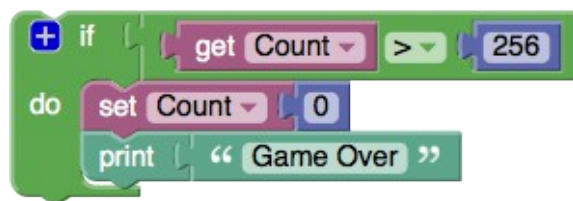


Figura 1: Editor Blockly. Fonte: Blockly homepage[7]

Os componentes de hardware seriam primeiramente leds e botões. Uma vez definidos e documentados na aplicação a mesma lógica poderia ser aprimorada em componentes mais complexos como sensores e motores. Todos os componentes devem acompanhar descrições de como devem ser conectados à *breadboard* e às portas físicas, de maneira a definir um padrão para componentes mais complexos.

A aplicação deve ser amplamente documentada para apoiar o desenvolvimento de novas funcionalidades.

3. Resultados Esperados

A conclusão do projeto depende do atendimento dos seguintes requisitos:

- Oferecer uma interface *web* que permita ao usuário definir rotinas de instruções arrastando e organizando elementos gráficos.
- As rotinas devem ser interpretadas pelo controlador, executando as instruções e

alterando o estado das portas físicas, refletindo nos componentes conectados ao controlador.

- Os componentes e a maneira de conectá-los devem estar documentados e disponíveis no repositório da aplicação.
- Afim de permitir a construção de um pequeno sistema de robótica, os componentes de hardware devem consistir em leds, botões, motores de corrente direta e sensores de proximidade.

Entre junho e setembro de 2014 o Raspberry Pi necessário para os testes foi adquirido. Afim de demonstrar a aplicabilidade da linguagem e bibliotecas escolhidas, foram realizados testes em que, usando Javascript, Node.js e a biblioteca Onoff, o Raspberry Pi foi programado para mudar o estado de uma porta física específica sempre que percebesse uma alteração de estado em outra porta específica, representando o funcionamento de um led que acendesse sempre que um botão fosse pressionado.

4. Considerações Finais

Os desafios do desenvolvimento do projeto, entretanto, serão a definição e o desenvolvimento das primeiras classes, a integração com Blockly na interface, e a documentação do projeto. Os autores ainda consideram a utilização de técnicas de *desenvolvimento orientado a testes*, afim de garantir a documentação e a integridade da aplicação.

É importante ressaltar que o principal objetivo do projeto é definir as bases de uma ferramenta de desenvolvimento contínuo. Dessa forma os autores esperam que o conjunto de capacidades da aplicação esteja cada vez mais próximo de utilizar as possibilidades oferecidas pelo ambiente. A programação orientada a eventos do Node.js poderia ser utilizada para programar o conjunto para executar uma rotina assim que recebesse o sinal de um sensor. Espera-se também que construção de componentes mais complexos leve conseqüentemente à blocos gráficos que representem conceitos de mais alto nível, como “mover adiante”, ou “pegar objeto”.

Agradecimentos

À Bruna do Carmo da Escola Superior de Educação e Comunicação, da Universidade do Algarve, pela colaboração.

Referências Bibliográficas

[1] The Raspberry Pi Foundation, (2013). Raspberry Pi. Disponível em: <<http://www.raspberrypi.org/>>. Acesso em: 11/9/2014.

[2] GROUT, V.; HOULDEN, N. Taking Computer Science and Programming into Schools: The Glyndŵr/BCS Turing Project. **Procedia - Social and Behavioral Sciences**, v. 141, p. 680–685, 2014. Elsevier B.V. Disponível em:

<<http://linkinghub.elsevier.com/retrieve/pii/S1877042814035435>>. Acesso em: 7/9/2014.

[3] SALEIRO, M.; CARMO, B.; RODRIGUES, J. M. F.; BUF, J. M. H. DU. A low-cost classroom-oriented educational robotics system. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**. v. 8239 LNAI, p.74–83, 2013.

[4] Code.org (2013). Disponível em: <<http://code.org/>>. Acesso em 11/9/2014.

[5] CAS, Computing at School (2013). Computing for the next Generation. Disponível em: <<http://www.computingatschool.org.uk/>>. Acesso em: 11/9/2014.

[6] Node.js Disponível em: <<http://nodejs.org/>>. Acesso em 11/9/2014.

[7] Blockly, A visual programming editor. Disponível em: <<https://code.google.com/p/blockly/>>. Acesso em 11/9/2014.

[8] Onoff. Gpio access and interrupt detection with javascript. Disponível em: <<https://github.com/fivdi/onoff>>. Acesso em 11/9/2014.

[9] Raspbian. Disponível em: <<http://www.raspbian.org/>>. Acesso em 11/9/2014.